AD-778 673

NATURAL COMMUNICATION WITH COMPUTERS IV.

Bolt Beranek and Newman, Incorporated

Prepared for:

Advanced Research Projects Agency

May 1974

DISTRIBUTED BY:

NTIS
National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

Best Available Copy

AD 778673

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Bolt Beranek and Newman Inc. 50 Moulton Street Cambridge, Massachusetts 02138	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

Natural Communication with Computers IV

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Quarterly Progress Report #14, 1 January 1974 to 31 March 1974

5. AUTHOR(S) (First name, middle initial, last name)

6. REPORT DATE May 1974	7a. TOTAL NO OF PAGES	7b. NO OF REFS
8a. CONTRACT OR GRANT NO DAHC-71-C-ØØ88 b. PROJECT NO c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) BBN Report No. 2822 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	

10. DISTRIBUTION STATEMENT

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

Under this contract, Natural Communications with Computers IV broad based computer science research and development work is performed in areas including; speech understanding systems, speech compression, development of programs and programming aids, techniques for extending computer I/O capabilities, research and development on time sharing systems, and distributed computation. This research program involves the ability to represent knowledge and deal with it in computer oriented terms, requires systems capable of a high degree of man-machine interaction, and draws upon many diverse fields such as linguistics, communications, programming, hardware development, speech recognition, etc. to facilitate fuller use of the enormous potential of natural modes of communication with computers.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

NONE

Security Classification

ia

NONE

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
time sharing systems						
speech understanding						
distributed computation						
programming aids						
BBN-LISP						
Interlisp						
TENEX						
ARPA Network						
Computers and communication						
resource sharing						
speech compression						
vocoders						

DD FORM 1 NOV 65 1473 (BACK)

S/N 0101-807-6821

Security Classification

A-31409

TABLE OF CONTENTS

I. INTRODUCTION

This report, our fourteenth Quarterly Progress Report on Natural Communication with Computers, covers our achievements from January 1, 1974 through March 31, 1974 in the areas of (a) Speech Understanding systems (b) Distributed Computation and TENEX-related activities (c) Languages and (d) Speech Compression.

In the preceding quarter (Fall, 1973), our Speech Understanding project passed a significant milestone in the November review by the speech Steering Committee. In contrast to the extensive programming and development activities leading up to the review, the past quarter has been spent in documenting the project status and progress to date as well as in design efforts preparing for the next round of development activity. In addition, a new problem domain has been selected (discourse about travel budgets) and the groundwork laid for incorporating this new subject domain into the current speech understanding system.

In our distributed computation and TENEX projects, the major emphasis this quarter has been upon the overall reliability of the TIP-Network-TENEX complex as perceived by the ultimate end user. While we are pleased to report other areas progress in these projects, the reliability effort has consumed a major amount of the available manpower resources. Other noteworthy activities include improvements in the area of TENEX security and participation in a number of planning meetings held during the quarter.

Our language related activity has continued its work on LISP-related enhancements in response to the needs of the research user community. An encouraging development during the quarter has been the establishment of an improved dialogue with the MIT LISP community resulting in an improved awareness of user requirements and an examination of ways to achieve greater system commonality.

During the past quarter, we have worked on various aspects of our speech compression system to produce good quality speech at low transmission rates. At the March meeting of the ARPA Network Speech Compression (NSC) group we demonstrated the transmission of good quality 10 kHz sampled speech at low rate of 2800 bits/sec. Since then we have developed encoding schemes that enable us to produce the same speech quality at lower rates, close to 2000 bits/sec. For the speech compression system demonstrated, we used the log area ratios for both encoding and interpolation, and performed a time-synchronous synthesis. We have also written up the results of our research on quantization of transmission parameters as a BBN Report (Makhoul and Viswanathan, 1974).

II. SPEECH UNDERSTANDING

A. Introduction

During the past quarter, much of our activity has been devoted to planning the next phase of development of our continuous speech understanding system, following the completion of our demonstration system in November 1973. The experiences of building that system and observing the interactions of its components has generated new ideas for the next step, and we have taken the opportunity to evaluate them.

Another significant activity during the last quarter was writing a set of papers giving a comprehensive description of the organization and operation of the BBN SPEECHLIS speech understanding system. For many components of the system, this was the first major description of them. These papers were submitted to the IEEE Symposium on Speech Recognition, to be held 15-19 April 1974, at Carnegie-Mellon University, and they will appear in the Proceedings:

1. "Motivation and Overview of BBN SPEECHLIS: An Experimental Prototype for Speech Understanding Research"
2. "Where the Phonemes Are: Dealing with Ambiguity in Acoustic-Phonetic Recognition"
3. "Where the Words Are: Lexical Retrieval in a Speech Understanding System"
4. "The Use of Syntax in a Speech Understanding System"

5. "Semantic Support for a Speech Understanding System"
6. "Control Concepts in a Speech Understanding System"

These papers have also been submitted to the IEEE Transactions on Acoustics, Speech, and Signal Processing.

B. New Task Domain

In the past quarter, we chose and began the construction of a second problem domain for the speech understanding project, travel budget management. Within this domain, a user will be able to phrase such requests as:

1. What trips did we have budgeted for the speech project as of September, 1973?
2. Which of them have already been taken?
3. Give me a list of the remaining trips with their estimated costs.
4. Nine people will be going to Pittsburgh in April for the IEEE conference.
5. The registration for that meeting is $40.
6. If we only send 3 people to London and 1 person to Stockholm, will we then be within the budget?
7. Change the cost of a trip to Amherst to $10.

That is, the user will be able to query the data base, add to it, and make both hypothetical and permanent changes to it.

There were many reasons for wanting to bring up a second domain and many things we feel we can accomplish within the new domain that would have been difficult or even impossible within that of lunar geology:

1. It has been very difficult for us to gain access to informants concerned with problems in lunar geology; hence to build a user model, discourse model, or problem-solving procedure for this domain would involve an enormous effort which would be completely off the track from the problems of speech understanding. Within BBN, everyone is to some degree concerned with managing travel budgets, and there will be ample opportunity to observe people going about problem-solving tasks with the system as tool. (In fact, until the new system is running, we will continue to use the technique of incremental simulation to gather user-system dialogues to guide us in building user and discourse models).

2. From a phonological point of view, there are too many "strange" and unfamiliar words in the lunar geology vocabulary. The travel budget domain will allow us to collect (and construct) a large body of sentences that are easily and freely spoken.

3. From syntactic and semantic points of view, the new domain affords many interesting problems that are not likely to appear in the lunar geology domain, such as hypothetical questions like sentence 6 above. Maintaining two different domains gives us the opportunity to observe the ability of our syntax and semantics to deal with all sorts of concepts and

constructions.

4. The travel budget domain will enable us to demonstrate a system that is easily comprehended by a lay audience, something which lunar geology did not. This seems to be a worthwhile ability.

Thus far, we have constructed a small vocabulary on the order of 325 words for the new domain, complete with phonemic and syntactic properties, and we are in the process of building a semantic network to represent their meanings and likely contexts. We have also spent time in designing a retrieval language and data base for the system, work which will be continued in the next quarter.

C. Signal Processing

In January we attended a meeting of ARPA contractors concerned with signal processing hardware for speech applications. At this meeting and in a subsequent conference call, we decided on a standard signal processor configuration (Signal Processing Systems Inc. SPS41 processor, 8K of bulk memory, PDP11/40) and formed an ARPA SPS41 user's group. Since then we (BBN) have ordered an SPS41 processor and have been selecting PDP-11 peripherals for a complete PDP11/SPS41/real-time-interface speech processing system which will be accessible from the ARPANET.

D. Segmentation and Labeling

The major effort in segmentation and labeling during this quarter was spent in planning and implementing research tools for developing a more accurate Acoustic Phonetic Recognition (APR) program.

An interactive program has been implemented which allows an experimenter to do statistical studies over a data base of utterances of correlations between certain phonemic contexts and user-specified functions of parameters in that region. For instance, the user might ask for all occurrences of a word-initial unvoiced plosive followed by either a front vowel or a schwa. Then, on finding such a context, he might want to tabulate the minimum value of the energy in the unvoiced plosive, the average value of the normalized minimum error during the central half of the vowel, and the value of the energy in the preemphasized signal at its first local maximum which is at least 10 dB above and occurs after its minimum value within the plosive.

This facility will provide a method of determining quickly whether a particular algorithm would yield evidence about the existence of certain features. If the algorithm seems useful, this will provide a statistical distribution from which to compute the probability of the existence of the acoustic features, given the results of the algorithm.

E. Word Verification

Given the results of the Klatt-Stevens spectrogram reading experiment (BBN Report 2514), it seems clear that the ability to return to acoustic evidence for verifying word hypotheses is important to correct identification. This is because one can verify the consistency of all acoustic clues with respect to the given word hypothesis. Assuming that phonological and coarticulation processes are described by rules which are generative in nature, it is felt that an analysis-by-synthesis procedure is needed to overcome inaccuracies present in preliminary phonetic analysis and to decode the effects of the phonological rules. The synthesis component of a word verification mechanism must be able to transform an abstract word representation into an acoustic representation suitable for a comparison with the acoustic parameterization of an utterance.

Using a terminal analog model of speech production, one does a direct phonetic-to-acoustic parameter conversion using rules derived from relevant data collected from spectrograms or extracted automatically from digitized natural speech. In addition, it is possible to discover and quantify phonetic-to-parametric rules. A synthesis program has been written which generates three formant frequencies from hand-generated phonemic strings. Concurrently, a mapping strategy for comparing against unknown parameterizations is under development. This includes time registration, frequency and time normalization, and match score computation. Ultimately, a control strategy which evaluates match scores and obtains new word hypotheses must interface to the total speech understanding system. Thus, the specification of an analysis-by-synthesis approach will evolve as the separate components (synthesis, mapping and control) become operational.

F. Phonological Research

In January, we attended the project-wide Phonological Rules Workshop held in Santa Barbara, where we presented the implementation of analytic phonological rules in SPEECHLIS. At this workshop, a phonological rules review committee was formed. We are participating in the work of this committee in assessing rules contributed from seven sites. An initial presentation of this work will be given at the IEEE Symposium on Automatic Speech Recognition in April.

The word-verification work described above has led to the compilation of a new dictionary representing information on a phonetic level for the synthesis procedure. Particularly germane to the other phonological research was the matter of syllable boundaries examined and formulated in this new dictionary.

G. Syntax

Work has proceeded on the design of several new features to be incorporated in the syntactic component, principally a better handling of fuzzy word matches and the inclusion of register information in the partial syntactic parse paths which are constructed by the system. This version of the parser allows for more merging of transitions and sharing of register data so that information can be shared more efficiently within and among various theories. We have also begun to work with Wayne Lea at UNIVAC and Mike O'Malley at Berkeley to determine what prosodic information can be made available to the system and how the syntactic component can take advantage of it.

III. DISTRIBUTED COMPUTATION AND TENEX RELATED ACTIVITIES

A. Introduction

This quarter, we started to exploit another service capability made possible by the RSEXEC distributed computation environment: a multiplexed message service. A user of this service will be able to read up-to-date versions of his mail file through any cooperating TENEX host. This is made possible by regular, periodic interaction of the various RSEXEC servers to exchange and update redundant copies of user message files.

Service facilities (FTP,TIPSER,RSEXEC, etc.) will soon take advantage of a new system capability which was implemented this quarter. Each instance of service provided by a different process can now be provided in a separate security protection domain. Use of this facility will insure that TENEX security cannot be compromised by errors in the service facilities.

The reliability and accessibility of TENEX as experienced by TIP users have been greatly enhanced through the efforts of a special joint project of the BBN-TENEX group and the BBN-TIP group. Network connections are now maintained (with informative messages to the user) between the BBN-TESTIP and the BBN-TENEX through TENEX service interruptions. These improved systems will soon be distributed to all TIP and TENEX sites.

The security of the TENEX monitor has also been improved by standardizing the code which handles passwords, and by maintaining a threat log of jobs which have excessive rates of password failure. When the failure rate exceeds a certain threshold, the operator and user are notified and the job is summarily logged out.

B. Meetings

We attended a Packet Radio meeting in Hawaii on January 11. The purpose of the meeting was to familiarize all participants with the current state of the packet radio project and to coordinate the various development efforts.

We participated in a three-day meeting of the ANTS Steering Committee at the University of Illinois, January 14 - 17. The purpose of the committee meeting was to evaluate the general usefulness to the ARPA community of the ANTS development effort. It resulted in specific recommendations to the ARPA office about the priority of ANTS development tasks and the level of support needed to carry them out.

BBN hosted a Management Systems Technology meeting for all interested ARPA contractors January 31 and February 1. The purpose of this meeting was to mobilize a research, development, and technology transfer effort in this area as part of a new five-year ARPA-IPT program. The result of this meeting was the creation of an MST Advisory Committee to arrive at a scientific plan for programmed creation of professional management tools.

We also attended a TENEX meeting at the ARPA office on February 25 and 26. The purpose of this meeting was to evaluate the current and future TENEX support requirements of the various ARPA research projects, and consider means of meeting these requirements. Areas of special interest were requirements for development of new system facilities. requirements for computing power, and the problems of system standardization with multi-site system development efforts. The result of the meeting was the formation of a TENEX Advisory Committee to evaluate the above requirements and make recommendations to ARPA with respect to priorities and resource allocation.

On March 14 and 15, we attended a Front End meeting at the ARPA office to discuss issues related to network access mini-hosts, or "front-ends". Several significant decisions were made at that meeting: the ANTS Steering Committee was reconstituted as the ELF Advisory Committee and charged with surveying the requirements of potential ELF users, and recommending the most useful areas for ARPA support. A decision was made to standardize the command languages provided by ANTS, ELF, and TIPSER/RSEXEC into subsets of a single command language, permitting users to access the network in a uniform manner through any of these access systems. An interface standardization committee was also formed to evaluate the four existing PDP-11 to IMP local interfaces, and specify a standard interface which best meets the needs of the ARPA community. We will be playing an active part in the operation of this committee.

C. Distributed Computation

1. Multiplexed Message Service

As part of the effort undertaken by the BBN-TENEX and BBN-NET groups to improve the reliability and quality of network and host services, we have begun development of a "coupled" message service. The basic idea is to take advantage of the redundancy represented by ARPANET TENEX hosts to provide a highly accessible network message service and to make the service available to users in a convenient, host independent way. The service is based on cooperative operation of TENEX hosts in which the hosts interact on a regular, periodic basis to exchange and update user message files. As a result, the user will be able to read an up-to-date version of his message file through any operating host.

----

*The document is on-line at BBN as file <DOCUMENTATION>COUPLED-MESSAGE-SERVICE.DOC.

To implement this service we are applying techniques that we developed to provide the TIP-RSEXEC network executive service. Before we started an implementation we prepared a detailed specification of the coupled message service which we distributed for review to the ARPA office, to representatives of the TENEX sites at the ARPA TENEX meeting, to attendees of the ARPA Front End meeting and to other interested parties.* The following discussion assumes familiarity with that document.

The current status of the various components of the message service is as follows:

a. User access to the service will be provided through an extension to the existing TIP-RSEXEC system. The extensions required to the existing, operational system are minimal. They will be to make READMAIL and RD available to users after appropriate access control checks are made. (See d below).

b. Automatic backup of message files will be provided by an extension to the mechanism used to maintain multiple images of the TIPNEWS file at the TIP-RSEXEC sites. This extension has been designed and its implementation is scheduled to begin shortly. The data exchange protocols are designed to insure that messages previously read at one host need not be read in a subsequent message session on another host.

c. To provide an adequate environment for the message system software, modifications to the TENEX monitor were required. These modifications, which are described in more detail in the next section, have been designed, implemented and checked out and are scheduled for distribution as part of version 1.32 of the TENEX monitor.

d. We have asked the RISOS group to review our design specification with an eye for possible security problems. In the initial version of the service a user will be required to pass two access control checks to read his message file. First, he will identify himself by name and his "message service" password and then he will be required to supply his (login) password for the host providing the service. While this two step authentication procedure may seem somewhat excessive, we feel it is important that the multiplexed message service not compromise the privacy of users' message files. The two step check guarantees that access to a user's messages via the service is controlled as tightly as access via standard login. Subsequent implementations of the service may relax the access control checks as we come to understand better the access control requirements of our users.

2. Improved Execution Environment for Network Service Processes

TENEX provides the standard network services to remote users by dedicating a detached job logged in as user SYSTEM to each service (e.g., FTP, RSSER, TIP-RSEXEC). Each instance of a particular service, for example the TIP-RSEXEC service provided to a user of the AMES-TIP, is provided by a separate process (or group of processes) within the job. A single process within the job is dedicated to creating such process groups in response to (initial) remote requests for service.

While this approach has been adequate for providing the standard services, two aspects of TENEX have prevented it from being used to grant access to a wider range of services:

a. The access control provided by the monitor is currently on a "per job" rather than a "per process" basis. As a result, different process groups within a job supplying service to different remote users are subject to identical access controls rather than separate controls specific to each of the different users. Furthermore, the access controls in effect are based on the job's login name (SYSTEM) rather than on the identity of the remote user. (For a more detailed explanation of this, see previous QPR, BBN Report #2607). Thus to avoid compromising TENEX security each service process must run in a privileged mode in order to simulate the standard TENEX access control mechanism in effect for its particular user. In addition to requiring duplication of programming effort, the necessity of simulating TENEX access control increases the likelihood of security cracks due to programmer error and precludes the possibility of making user supplied software accessible through service jobs.

b. Terminal interrupts (i.e., EXEC ↑C, READMAIL RUBOUT) can originate only from the job "controlling" terminal. Because a service job runs detached, there is no such terminal, and even if there were one, a single terminal would be insufficient for multiple instances of service. Thus, each process that wishes to provide terminal interrupt capability to remote users must simulate it.

We felt that simulation of access control and terminal interrupts was inadequate for the message service described above. In particular, use of simulation would require rewrites of SNDMSG, READMAIL and RD and any other subsystems to be made accessible through service jobs in the future.

To provide a more satisfactory execution environment for service processes, we have generalized the access control and terminal interrupt features of TENEX:

a. The TENEX connected directory concept has been modified to permit separate processes within a job to have their access controlled differently. A process and its inferiors can be designated as a "fork group" for purposes of the CNDIR (Connect Directory) JSYS. The effect of execution of CNDIR by a process is limited to processes in its group. In addition, an option has been added to CNDIR which results in a change to the "login" directory as well as the connected directory. This permits the access granted to "service" processes to be based on the identity of the remote user rather than on user SYSTEM.

b. The terminal PSI concept has been modified to allow more than one source of terminal interrupts in a job. An assigned terminal can be designated as the source of terminal PSI's for a fork and its inferiors. This change represents a slight generalization of the controlling terminal concept: each process in a job still has at most one source of terminal PSI's but different processes may now have different sources.

3. Responsiveness of the Broadcast ICP

We have previously observed that the TIP @n command which used a "broadcast ICP" (see previous QPR BBN Report #2607) to access a TIP-RSEXEC is extremely unresponsive at times. We had attributed this slow response to heavy loads at the TIP-RSEXEC sites. Because we plan to use the broadcast ICP as a means for providing users site independent access to new network services such as the coupled message service, we spent some time this quarter improving its responsiveness. In doing so we discovered a fundamental flaw in its current implementation that accounts for its poor responsiveness (and a relatively easy way to improve the situation).

The broadcast ICP is a very simple mechanism. To establish connection with a service, the requesting party (usually a TIP) bids for the service by broadcasting requests for connection (RFC's) to all sites which may provide the service. It accepts the first site to respond and rejects all others.

In the current implementation a TIP does not remember to whom it has broadcast RFC's and will accept the first "reasonable" RFC that is returned to it to establish connections with a TIP-RSEXEC. Because it has not remembered to whom RFC's have been broadcast, the TIP treats the matching CLS's (corresponding to the broadcast logger socket and required by the Host-Host protocol) and the extra RFC's (from the slower TIP-RSEXEC sites) as unsolicited protocol commands. Due to extremely limited space for buffering such "unsolicited" commands (room for two such commands for the entire TIP at any time) the TIP cannot respond properly to all of them. The TENEX NCP, on the other hand, expects responses to the CLS's and RFC's and will not allow the corresponding sockets to be reused until it receives them or time outs occur (approximately two minutes).

There are currently two sites providing TIP-RSEXEC service. When both sites are up a TIP will receive two CLS's and four RFC's from the two sites in response to an @n command. Both CLS's and two of the RFC's will be treated as unsolicited. Thus, use of the broadcast ICP mechanism, which was designed to provide quick access to a TIP-RSEXEC, will almost certainly "hang" the logger socket at one and quite possibly both of the TIP-RSEXEC sites with the result that subsequent connection attempts made within the two minute time period will experience very slow responsiveness. It is interesting to note that increasing the number of sites providing the service will not improve this situation. Rather, it will merely insure that more of the logger sockets must be timed out before they can be recycled.

The solution to this problem is to have TENEX "violate" the host-host protocol also by allowing the TIP-RSEXEC logger sockets to be recycled without waiting for the (never to come) CLS's from the TIP. The necessary changes to the TENEX NCP have been made and will be distributed as part of version 1.32 of the monitor.

The broadcast ICP is a good example of an application for which the current ARPANET Host-Host protocol is much too complex. To use the broadcast ICP the requesting party need only signal each server that it requires service and then wait for the fastest to respond. Unfortunately, in order to do this in accordance with Host-Host protocol it must participate in an elaborate exchange of protocol commands with each of the servers, carefully remembering the state of each exchange. For large hosts this exchange is only wasteful, for small hosts it is impossible.

4. RSEXEC Development

During this quarter the RSEXEC system was installed at the OFFICE-1 TENEX. In addition, we have made the source files for RSEXEC available to the XEROX PARC-MAXC host which has just begun running version 1.31 of TENEX. We expect (a perhaps limited version of) RSEXEC to be installed on the PARC TENEX shortly.

The question of the load placed on TENEX by RSSER (the RSEXEC server program) arose at the recent ARPA TENEX meeting. Several widely varying claims were made including our own estimate that RSSER uses about 1% of the CPU. Fortunately, each RSSER keeps a comprehensive log of its activity. An examination of the log files for all RSSER sites corroborated our estimate. Given the current high demand for computing resources and the fact that not all of the CPU is available to users due to system overhead, the 1% figure must be considered too high. A careful examination of the RSSER program revealed that the time constant associated with the TENEX status exchange function was much too small. A new version of RSSER with a much larger time constant is now running at all TENEX sites. As a result, the load on TENEX due to RSSER has been reduced to a more acceptable .3 - .5% of the CPU.

At the ARPA Front End meeting a committee representing the ANTS, ELF and TIP-RSEXEC systems was formed to design a standard command language for these systems. The committee recognizes that the TIP, ELF and ANTS differ in philosophy, emphases and capability. The goal of the committee is not to emasculate the systems by suppressing their differences. Rather it is to specify a uniform command language structure (e.g. standard ways to invoke "help" features, to report error conditions, to invoke command recognition and completion, etc.) and where identical capabilities are supported to specify identical command names for invoking them. The benefit to the user is clear: he need learn only a single language to use all three systems; where the systems differ he can concentrate directly on their significant differences rather than on the superficial differences in their command languages. We have prepared an initial design specification for the language which is being evaluated by the committee.

The RISOS Project at Lawrence Livermore Laboratory has obtained the source files for the RSEXEC system in order to evaluate it in terms of possible compromises to TENEX security it might represent. We regard RSEXEC as an important part of ARPANET TENEX and we await their report with great interest.

D. TENEX Improvements

Improved Reliability and Accessibility

a. TIP/TENEX Reliability Improvement

At the request of the ARPA office, we instituted a special project this quarter to improve the reliability and accessibility of TENEX as observed by a TIP user. This project was a joint effort by the BBN-TENEX group and the BBN-TIP group, and involved extensive and cooperative modification of both systems. The first step in this project was to catalog and diagnose all known reliability problems, and then to prepare a formal project plan (for review by the ARPA office) describing the steps necessary to correct these problems.

In accordance with the plan approved by ARPA, we have taken the following steps:

1. TENEX now automatically retransmits all messages which receive "incomplete transmission".
2. The "IMP going down" message is interpreted and typed out to all users.
3. On-line status of network service disruptions is now available through the IMPSTAT command to the EXEC.
4. Users are now permitted to ATTACH to existing jobs in spite of DRUM, SPT, or JOBS resource limitations.
5. A user reattaching to a job can no longer go into a hung state due to process PBIN from a disconnected network virtual terminal.
6. All waiting output is now flushed on receiving CLS, avoiding long close timeouts.

7. Extensive validity checking is done on requests for connection. Duplicate connections are detected and the first one is closed before opening the second.

8. The IMP driver has been restructured to maintain the NCP state across a drop of the ready line: only a reinitialization of the interface results.

9. The extended host-host protocol is used to resynchronize connection allocation after service interruption. For the first time, TIP-TENEX connections are maintained across TENEX service interruptions.

These improvements, along with several others, will be distributed in TENEX system 1.32 and matching TIP system 322.

b. Core Dump

A facility has been added to TENEX to permit non-destructive dumping of core images to a special disk area. An image can be retrieved by a new subsystem, GDUMP, which copies it onto a SSAVE file. This file can then be inspected with IDDT The use of this facility in no way interferes with the possibility of resuming operation of a crashed system; its intent is to improve system availability by eliminating certain instances of protracted on-line debugging and by allowing more intensive investigation into the causes of system failures.

c. Bugnote

A third category of system error has been added, called "bug notes". These are correctable errors of lesser severity than "bug checks" and never cause the system to halt; they are simply noted on the JOB Ø teletype. This category was added to eliminate unnecessary interruptions.

2. Improvements in Resource Management

a. Memory management change

A change has been made in the way TENEX accounts for memory committed to processes. This change avoids double-accounting for shared and/or locked pages, which was an unfortunate consequence of the previous algorithm.

b. Balance set management change

The scheduler's balance set management policy has been changed because it was recognized that the previous policy caused the system to appear excessively "sticky" under heavy load. The new policy allows processes into the balance set if (a) memory is not over-committed or (b) space can be made for the candidate process by displacing lower priority processes currently in the balance set.

If a process achieves entrance to the balance set via criterion (b), the lower priority processes are not actually removed from the balance set until an actual need for their space developes. Thus, incorrectly large space-requirement estimates for candidate processes do not cause unnecessary process-thrashing.

3. Security Improvements

As part of our continuing effort to improve the security of the TENEX monitor, the instances where user passwords are checked were reviewed and made uniform. Additionally, a check is now performed to detect jobs having excessive password failure rates. When the failure rate exceeds a certain parameter, the operator is informed of the fact and the job is logged out. This prevents users from attempting to discover passwords by exhaustively testing all possible passwords.

4. Peripheral Processor

a. ELF Operating System

In order to support remote TENEX peripherals on a network mini-host, we installed the ELF PDP-11 operating system developed by the Speech Communications Research Lab, Santa Barbara, on our BBN-11X system (Host 5). This required two steps: first, we extended our PAL11X cross-assembler to assemble the MACY11 source code of ELF; this required the addition of a second symbol table to PAL11X. Secondly, we created a clock driver module for ELF which is compatible with our KW-11L line clock. The ELF system came up easily and performed well, providing both user TELNET and FTP server facilities. The FTP server was used as a data path for driving our line printer across the network from TENEX.

b. New LPT spooler and driver

A recent paper by Cerf and Kahn [1] describes a new host-host protocol with a flexible mechanism for achieving network message flow control and error recovery. We have begun an experiment to investigate the properties of this protocol and, if necessary, to produce an improved version. The application chosen for the experiment was spooling line printer listings from BBN system A to the BBN-11X line printer. The new protocol was implemented using the network raw message facility provided by TENEX.

The initial implementation was a subset of the complete protocol, containing the retransmission and resynchronizing procedures, but lacking the ability to have the receiver control the window size used by the sender. Missing also was a way to establish and dissolve associations. The former is straight-forward and is included in the second version of the software (TENEX spooler and 11X printer driver) which is under development.

We feel that significant changes to the protocol will be required in order to permit proper establishing and dissolving of associations. The main problem is the "duplicate message" problem in which a receiver may be restarted and request synchonization from the sender, but in fact may see synchronizing information which was queued in the network from before it was restarted. This of course will be confused with the information which it actually requested.

The initial version of the spooler was run in a semi-production mode (producing listings for the TENEX group) for approximately two weeks. During this time the duplicate message problem appeared several times.

We also feel that a method must be devised to enable the receiver to state that it is refusing to establish an association. This is preferable to letting the sender(s) repeatably retransmit until a positive acknowledgement is received.

5. Direct Network Message Facility

A facility has been added to TENEX to permit users with a special capability to send and receive messages directly to and from the ARPA network. This permits experimental protocols to be evaluated in user code without interfering with normal network traffic. The facility enables a user job to specify that all messages having a particular header value when masked with a particular mask be directed to his job. Furthermore, only his job can send messages with the same header values. To guarantee that one job cannot interfere with another's use of this facility the system prohibits overlapping assignments. Provision is made for 8 different value/mask queues. The facility is currently in use by the IMP group for monitoring certain aspects of network performance and for our experiments with the Kahn-Cerf protocol.

6. New TELNET Protocol

a. General Implementation of TELNET User Primitives

A package of routines implementing all of the primitive operations needed for a user TELNET program (making connections, network I/O, negotiating options) has been prepared. It is written in BCPL and hence can (hopefully) be used for TELNET programs on various computers. Included in the package (called TELIRM) is a data structure describing the storage of all data specific to a given network connection, such as JFN or stream discriptors, states and negotiation statuses of all TELNET options, and I/O interlocks. Among the features of the package:

1. All routines take a connection number as an argument (used as a subscript on the connection data), so the package can be used by programs that serve multiple

connections.

2. The routines handle all the details of negotiation of TELNET options initiated by either party of a connection: checking legality, sending commands and responses, setting option states. At the completion of each negotiation, the package calls specific action routines in the controlling program.

3. Although the controlling program must provide the memory allocation for the TELPRM data block, it need never, and should never reference the connection-specific data directly. Routines are included to return specific data values if needed.

b. New User TELNET

The primitive package has been checked out and successfully installed in a new version of the User TELNET (presently operating on BBN-TENEX as NTELNET). This "field test" version currently negotiates its connections via the foreign host's socket 27 (octal), a temporary Network convention for protocol connections. Three new-protocol options are now implemented: (remote) Echo, Suppress-Go-Ahead, and Timing-Mark. Implementation of Transmit-Binary and RCTE (Remote- Controlled Transmission and Echoing) are planned for the near future.

c. New TELNET Server

The TENEX monitor now implements the new server TELNET protocol in addition to the old protocol. The selection of which protocol is used is on the basis of the value of an argument to the ATPTY JSYS which is used to attach a pair of network connections to an NVT. The program which responds to requests for connection to an NVT (NETSER) in turn sets the flag when responding to requests for connection arriving on socket 23(decimal). Currently, the options implemented are:

Suppress Go-Ahead. Both directions are implemented. An initial request is made to suppress GA's in both directions.

Echo (by server). This option is negotiated in response to the execution of an STPAR JSYS which changes the setting of the full/half duplex bit. Failure of the option negotiation nullifies the attempt to change modes. The STPAR JSYS is done in response to the EXEC's HALF and FULL commands.

Binary mode. Can only be initiated by doing an SFMOD JSYS specifying binary mode. Both directions are negotiated into binary mode and if either fails, the other is retracted.

Timing Mark. This is generated by the DOBE JSYS. DOBE dismisses until both the local terminal buffers and the network pipe are emptied.

The GA character is transmitted (when not suppressed) whenever a fork blocks for terminal input or whenever a fork which was previously waiting for terminal input is unfrozen (RFORK JSYS). This is the best that can be done and will not work when multiple forks are using the same terminal for input and output.

7. Mail System Improvements

A number of improvements have been made in the reliability, error handling, and user interface of the programs which make up the TENEX mail system. The versions of SNDMSG, MAILER, READMAIL, RD, and MAILSTAT described below have been made available to all TENEX sites.

a. SNDMSG

Many bugs reported by other sites have been corrected, and more informative error messages have been installed. Control-O has been added as an interrupt character which supresses typeout. This is useful, for example, for aborting a help message after the point of interest. The processing of the user list has been changed to provide more sensible defaults and explanations. The format of mail headings has been changed to conform to the network standard mail heading specified in RFC #561. This rework of SNDMSG will be continued into next quarter, during which we will extend the user's command language to permit a carbon copy list and editting of the message being composed.

b. MAILER

MAILER is the autonomous system process which periodically delivers queued mail. It has been modified to make much better decisions about which failures cause the mail to be undeliverable, and which failures are temporary, requiring requeing of the mail. The error messages returned to the user for undeliverable mail are now more informative, and several problems, which caused either multiple delivery of messages or complete loss of messages, have now been corrected. Of course, the header format has also been changed to conform to SNDMSG and RFC #561.

MAILER can now be invoked by any user to attempt transmission of his queued mail The final disposition of each queued message is reported to the user.

c. READMAIL

A number of bugs reported by other sites have been corrected, and error messages have been made far more informative. The default start date for both MESSAGE.TXT files and user specified files is now the date of last reading: all messages appended since that time are printed.

d. RD

RD has been updated to process the new network standard mail header specified in RFC #561, in addition to the old mail header. However, in understanding a wider variety of mesage headings, the new RD takes noticeably longer to run than the old one did. This problem will be cured by i the same facilities into an efficient programming language: the present RD is written as a set of TECO macros which execute interpretively, resulting in very poor CPU efficiency.

e. MAILSTAT

MAILSTAT (MAIL STATus) is a new subsystem for handling undelivered mail. It lists all queued and undeliverable mail in the connected directory. It also accepts commands to manipulate the undeliverable messages - they can be deleted or put back on the queue to be mailed (with a different address if desired). These functions were previously difficult to accomplish by other means (directory listings, deletion, renaming) due to unusual characters used in the file names of queued and undeliverable mail.

8. BSYS

The TENEX Backup System (BSYS) has been extensivly modified in several areas: improved error recovery, correction of operational deficiencies, ability to operate on TENEX systems with more than 1000 directories, ability to save and restore file page protection, proper handling of file accounts, ability to permit file protection to be a string, and correct restoration of files which have special characters (for instance at-sign as in undelivered mail files).

Coupled with the BSYS activity, we have implemented special files, called "undeleteable" or "perpetual" files. These files as immune to the DELF (delete file) JSYS and are used for files such as the system disk bit table and users' archive directories.

9. EXEC

The EXEC has been extended to interface with the new BSYS. A memorandum describing these changes has been distributed to all users of BBN TENEX.

A command, "IMPSTAT" has been implemented in order to provide users with information concerning TENEX-IMP reliability related events such as whether or not the two are currently talking to each other and the exact time and date of the most recent time TENEX reset its internal network tables. IMPSTAT will be expanded as additional information becomes available.

Upon initial startup for a new job, the EXEC now types a warning message if some resource such as drum space is low enough to prohibit additional LOGIN's. Additionally, users are advised that ATTACH'ing to existing jobs is permitted.

In order to cure an existing bug, and to provide a proper interface with the new fork group JSYS's, the EXEC now saves the identity of its primary input and output files at start up time. These are restored at the occurrence of an error condition, end of file on primary input file, of control-C terminal interrupt.

Additionally, some of the hooks for the SRI-ARC group allocation code have been installed.

10. TECO

A new entry point has been added to the TECO text editor in order to provide a smooth interface with the SNDMSG mail system. Currently, this simply returns a code describing the format of the text buffer, and pointers into

TECO's address space which tell SNDMSG where the actual text buffer is located.

11. LOADER

The LOADER has been modified to permit setting of a standard TENEX entry vector for assembly language programs ending with the "END LENGTH,,START" pseudo instruction.

12. IDDT

A pass has been made through the IDDT debugger in order to remove several existing bugs. Two new features have been added: the Control-T interrupt handler and the ;? command. The Control-T interrupt is handled much as the EXEC does but has the advantage that times are printed in seconds, the location where the user program is running is printed symbolically, and in addition to the load average type out, an "activity ratio" is printed. The activity ratio is the ratio of the console time to CPU time and serves to provide users with an indication of what fraction of the machine they are actually receiving. The command ";?" types a string which explains the most recent error condition in the program being debugged. If a specific argument is given (a standard TENEX error designator), the error string corresponding to that condition will be typed.

13. TAPET

A user mode magnetic tape test and diagnostic program has been written. Although it was originally planned for DEC TU30 tape drives (which several sites have), it has also been used with the STC drives being installed at BBN.

IV. LANGUAGES - LISP

A. Meeting

We met with several people at MIT to discuss the possibility of Interlisp satisfying the needs of current users of MAC LISP. In particular we discussed a list of about 16 desiderata. Some of these already were satisfied, some we are working on currently (e.g. read macros), some will be done in the future, some seem difficult or impossible while retaining all the features of Interlisp, and yet others (e.g. BIGNUMS) unlikely to occur without some help from MIT. The discussion has stimulated much thought about methods for achieving increased efficiency and about useful features to add.

B. Multiple Environments

The critical version of the compiler for multiple environments has been completed. This version contains all the regular compiler features, but lacks the block compiler.

C. Compiler

As part of the continuing cooperation with XEROX Parc, several improvements given us by Peter Deutsch were incorporated into the compiler. Mostly, these improvements involved extending the checking done to determine if an expression can be directly compiled into AC2 without disturbing AC1.

This, in turn, pointed out that this is one place in the compiler where further improvements are possible. One such improvement, which has now been made, is in the code generation for EQ's. A similar optimization is also made for FRPLACA and FRPLACD when the value of the expression is not needed.

D. Extended Read

An initial version of a new Lisp READ has been implemented. READ has been changed so that at no place does it check for a specific character. Instead, it maps the character into a table and checks to see if the table entry has the proper status bits set. Since the tables are implemented as regular Lisp arrays, the user can switch and manipulate these tables (called readtables) at will.

Facilities have also been provided to change the syntax meaning of individual characters in a given readtable. This allows the user to specify different characters to be used for parentheses, brackets, string delimiter, etc. By using several readtables, the user can switch quickly back and forth between completely different syntaxes.

In an attempt to give the user even more control over the input syntax, we have implemented read-macro characters. A read-macro character is a character which is declared by the user along with an associated function. Whenever READ sees a read-macro character, it calls its associated function. The value of the function is used by READ in several different ways, depending on the type of read-macro the character has been declared as. The value can simply be added to the list being read, spliced in as a segment, or even completely replace the whole list.

Read-macros also allow more compatibility with other Lisp systems. Read-macros are very often used in programs written at MIT in MAC Lisp (e.g., Planner, Conniver), and it has always been somewhat difficult to convert such programs to INTERLISP. With read-macros, such programs can be converted without special input routines or scanners.

V. SPEECH COMPRESSION

In our speech compression project, several advances have been made in the area of encoding of transmission parameters. These include an optimal solution to the problem of allocating a given number of bits among the various parameters, and procedures to encode the transmission parameters. The encoding procedures are found to reduce the transmission rate from 2800 bits/sec to 2000 bits/sec without causing any change in the resulting speech quality. We have also conducted synthesis experiments using different types of parameters for the interpolation at the synthesizer, such as autocorrelation coefficients, reflection coefficients and log area ratios. In some experiments we used interpolation across voiced/unvoiced boundaries. We found that time-synchronous updating of the synthesizer parameters results in a slightly better quality speech than the pitch-synchronous updating when time synchronous analysis is used. In these and many other experiments we have specifically observed that the change in the speech quality due to any one improvement in encoding, interpolation, etc., is most often not perceivable, but when several such improvements are added together there is a clearly perceivable improvement in speech quality. The inability to perceive small differences in speech quality has reinforced our concept that some form of objective evaluation of speech quality should be incorporated within the speech compression system to generate performance scores and hence enable us to make relative judgments of these differences.

Also in the last quarter, a paper entitled "Transmission Parameters for Linear Predictive Systems and their Quantization Properties" was presented at the 1974 Arden House Workshop on Digital Signal Processing (Viswanathan and Makhoul, 1974).

A. Encoding of Transmission Parameters

The emphasis of our research in the past quarter has been mainly in the encoding of transmission parameters. In our last QPR (BBN Report No. 2721) we reported that the reflection coefficients are the best set for use as transmission parameters. In order to determine an optimal way of quantizing the reflection coefficients we developed a method based on a spectral sensitivity analysis of the coefficients. We defined the spectral sensitivity measure:

$$\frac{\partial S'}{\partial k_i} = \lim_{\Delta k_i \to 0} \frac{1}{\Delta k_i} \log\left[\frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{P(k_i,\omega)}{P(k_i+\Delta k_i,\omega)}\right] d\omega$$

where P(.,) is the spectrum of the linear prediction filter. The quantity after the summation symbol in (1) gives the absolute deviation in spectral amplitudes due to a perturbation in the ith reflection coefficient k . From the results of the sensitivity analysis of the reflection coefficients using the spectral sensitivity measure (1), we showed that the optimal way of quantizing the reflection coefficients consists of first transforming them to the log area ratios:

$$g_i = \log \frac{1+k_i}{1-k_i}, \quad 1 \le i \le p,$$

and then quantizing these linearly. These comments provide the background necessary for the presentation of the results that follow.

1. Interpretation in terms of Pole Locations

While the spectral sensitivity measure given by (1) is useful in quantifying the overall deviation in the spectrum due to perturbations in the reflection coefficients or the log area ratios, it does not, however, explain corresponding deviations in the pole locations of the linear predictor. Much is known about how the accuracy of pole (or formant) locations affect speech quality. Therefore, it would be useful to examine the pole deviations due to quantization of the transmission parameters. For simplicity, we considered a 2-pole model and following [1] (Kitawaki and Itakura, 1973) we used plots of root loci to interpret the relative quantization properties of the reflection coefficients and the log area ratios in terms of pole locations of the linear prediction filter. For the ranges of poles which are common in 2-pole analysis of continuous speech, we found that quantization errors in the log area ratios lead to a smaller deviation in the position of the poles than that obtained by the quantization of the reflection coefficients, assuming the same number of quantization levels in both cases (Makhoul and Viswanathan, 1974).

Kitawaki and Itakura considered still other nonlinear mappings of the reflection coefficients but concluded that the log area ratios lead to the best overall quantization accuracy. Our results make the stronger statement that the log area ratios are actually optimal in the sense that they possess a flat or constant spectral sensitivity behavior.

2. Optimum Bit Allocation

We have used the spectral sensitivity measure for allocating a fixed number of bits among the various parameters. Let $q_1, q_2, \ldots, q_p$ be the parameters chosen for quantization. These may be the reflection coefficients or the log area ratios or any other set of parameters. Given a total number of bits for quantization, M, the problem is to distribute these M bits among the p parameters in some optimal manner. In terms of quantization levels, the above problem may be restated as the allocation of $N = 2^m$ levels among the p parameters. Therefore, we have:

$$\sum_{i=1}^{p} M_i = M \quad , \quad \sum_{i=1}^{p} N_i = N$$

$$N_i = 2^{M_i} \quad , \quad 1 \le i \le p$$

where N_i is the number of levels and M_i is the number of bits used for q_i. We have chosen to derive the optimal bit allocation by minimizing the maximum spectral deviation due to the linear quantization of $\{q_i\}$. This can be quantitatively stated as a simple problem in constrained minimization (Makhoul and Viswanathan, 1974). Omitting the details of derivation, we merely give the solution to this problem as follows:

$$\delta_i \frac{\delta S}{\delta q_i} = \delta_1 \frac{\delta S}{\delta q_1} \quad , \quad 2 \le i \le p,$$

where δ_i is the quantization step size for the parameters

q_i. If $\underline{q}_i$ and $\overline{q}_i$ are the lower and upper bounds for q_i, then

$$\delta_i = \frac{\overline{q}_i - \underline{q}_i}{N_i} ,$$

From (3), (4) and (5) one can easily compute the optimal values for N_i, $1 \leq i \leq p$. For the log area ratios, the sensitivity curve $\frac{\delta S}{\delta q}$ versus q is flat. So, (4) implies that all the quantization step sizes be the same, which is intuitively clear. For this case, we have found it convenient and useful to begin with a particular quantization step size. That step size automatically determines the total number of bits needed, as well as the maximum spectral deviation, which in turn determines the resulting speech quality. One can then study the change in speech quality as a function of only one variable, namely the quantization step size.

3. Use of Another Spectral Sensitivity Measure

Beside the spectral sensitivity measure (1), other types of sensitivity measures may also be used in the quantization study. Specifically we have investigated a measure which is similar to the total-squared error used for minimization in linear predictive analysis. The reasons for our choice of this measure are: a) it results in an all-pole model spectrum that is a good approximation to the envelope of the spectrum of input speech, and b) it is mathematically tractable, unlike the absolute error measure (1). The new spectral sensitivity measure that we have considered is given by

$$\frac{\partial S'}{\partial k_i} = \lim_{\Delta k_i \to 0} \frac{1}{\Delta k_i} \log \left[\frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{P(k_i,\omega)}{P(k_i+\Delta k_i,\omega)} d\omega \right]$$

where $P(k_i,\omega)$ and $P(k_i+\Delta k_i,\omega)$ are the power spectra of the unperturbed and perturbed linear predictors, and have the same total energy. The sensitivity measure (6) can be analytically evaluated (Makhoul and Viswanathan, 1974). We found that the resulting spectral sensitivity has the same general properties as the spectral sensitivity obtained by using the measure (1) and reported in our last QPR. The only difference between the two is the actual shape of the sensitivity curve. The optimal quantization scheme for the reflection coefficients using the measure (6) can be determined in the same way as we did using the measure (1). This optimal scheme is to quantize linearly the following transformed coefficients (we call these log error ratios):

$$g_i' = \log \frac{1}{1-k_i^2}, \quad 1 \le i \le p.$$

We have experimentally investigated the quantization properties of the log area ratios (2) and the log error ratios (7). Through informal listening tests we have found that the use of the log area ratios for quantization leads to a uniformly better speech quality than that obtained using the log error ratios.

4. Statistics on the Log Area Ratios

For the quantization of the log area ratios as well as for determining the optimal bit allocation strategy discussed above, we need to know the ranges of the different log area ratios g_i, $1 \leq i \leq p$. It is clear that both over-estimation and under-estimation of these ranges lead to quantization errors. For a set of 12 speech utterances that have been sampled at 10 kHz, we extracted, at a rate of 200 frames/sec, the log area ratios through the linear predictive analysis using p=12 and an analysis interval of 20 msec. The maximum and minimum values were found for each log area ratio, and the corresponding range was then determined by allowing some margin on both of these values. We have also computed and plotted the histograms of the individual log area ratios. These histograms may be used in the design of more sophisticated quantization schemes.

5. Further Encoding Procedures

Near the end of the last quarter we began working on encoding procedures that would permit the transmission of parameters at lower bit rates, without causing any change in the speech quality. These procedures make use of the probability distributions for each of the parameters. In our preliminary experiments we found that Huffman coding can be used profitably to reduce our 2900 bits/sec transmission rate to rates close to 2000 bits/sec.

B. Interpolation Study

We have conducted synthesis experiments using different sets of parameters for linear interpolation. These include reflection coefficients, log area ratios and autocorrelation coefficients of the all-pole filter. Stability of the filter is preserved under interpolation in all three cases. The different interpolation parameters result in slight differences in the spectrum of the linear prediction filter but the quality of the synthesized speech as judged from informal listening tests did not show any perceivable differences. We are currently in the process of developing some objective measures to enable us to identify the best set of parameters for interpolation. Meanwhile, we continue to use the log area ratios for interpolation in our experiments.

Thus far in all our experiments we performed interpolation of the filter parameters between adjacent frames only when both frames are either voiced or unvoiced. Recently, we did some synthesis experiments where interpolation was also carried out also across voiced/unvoiced boundaries. In these instances we did not find any perceivable change in speech quality from the cases where such an interpolation was not done. We feel that more experimentation is needed to reach a definitive conclusion.

C. Synthesis

1. Time-Synchronous versus Pitch-Synchronous Synthesis

A considerable part of our efforts in the last quarter was spent in the investigation of time-synchronous synthesis. The synthesis parameters: (gain and predictor coefficients) were interpolated and updated time-synchronously (every 5 msec). Of course, pitch was still interpolated and updated pitch-synchronously. We have found that speech quality improves when the synthesizer parameters are updated at a time corresponding to the time when they were extracted in the analysis. Thus, if time-synchronous analysis is used, time-synchronous synthesis should also be used. Our experiments show no discernable "buzz" which might be expected to arise from time-synchronous updating.

2. Word Lengths for ADC and DAC

In view of the minimum phase property of the synthesis filter, the synthesized speech invariably has a higher peak amplitude than the original speech. Thus, if the ADC at the transmitter and the DAC at the receiver use the same word length and if full dynamic range of the ADC is utilized, the synthesized speech samples have to be scaled down before passing them through the DAC. Such scaling poses problems in a real-time application. Further, the scaled synthesis sounds less loud than the original speech. Therefore, we have made a modification in our speech compression system in which the DAC has additional bits at the high end to allow for overflow from the synthesis. We now use a 9-bit ADC and a 12-bit DAC. Previously, the DAC also had 9-bit samples as input.

D. Low Bit-Rate Linear Predictive Vocoder

At the March meeting of the ARPA NSC group we demonstrated several synthesized speech utterances having a low transmission rate of 2800 bits/sec. Briefly, we describe here the linear predictive vocoder that we simulated in our TENEX system and used for this demo. The input speech is sharply low-pass filtered at 5 kHz and sampled at 10 kHz using a 9-bit ADC. The autocorrelation method is used for the analysis with 12 poles and an analysis interval of 20 msec. The method of center-clipping is used for extracting the pitch. The parameters (energy, pitch and log area ratios) are extracted, encoded and transmitted time-synchronously at a rate of 50 frames/sec. For the encoding of the log area ratios, we use the maximum and minimum values for the different log area ratios that were obtained in our statistical study (section A.4) and we adopt the oprimal bit allocation strategy reported above (section A.2). At the receiver, log area ratios are used for linear interpolation. Both pitch and energy are interpolated logarithmically. The input signal to the synthesizer consists of a sequence of pulses for voiced sounds or uniformly distributed white noise samples for fricated sounds. The parameters of the synthesizer are updated time-synchronously at a rate of 200 times/sec. The synthesized speech samples are passed through a 12-bit DAC and then a low-pass filter wih a sharp cut-off at 5 kHz.

As mentioned in section A.3, use of further encoding procedures in the above vocoder cuts the transmission rate down to 2000 bits/sec without altering the speech quality.

E. Development of a Signal Processing System

Our effort in the development of a signal processing system has been largely a matter of system definition and information exchange. In defining the system, we have considered the needs of both the speech compression project and the speech understanding project. The requirements of these projects indicate that the system will have three distinct purposes: (a) To function as a real-time data acquisition system, supporting ADC's and DAC's at sampling rates up to 20 kHz, and providing a means of storing and retrieving speech utterances; (b) To allow the implementation of real-time speech analysis and synthesis and to make possible the transfer of coded speech over the ARPA network; (c) To provide signal processing computational power to multiple users, functioning as a peripheral to TENEX and serving to remove some of the computing load from it.

In cooperation with the other sites involved in these two projects, we have been investigating various items of hardware and software with which to implement the system. This cooperation has resulted in the network-wide selection of the SPS-41 over both the SPS-81 and the CSPI-4001 as the signal processing computer, one of the most critical parts of the system. This cooperation has also given the sites involved a considerable leverage with the manufacturer, SPS, resulting in significant hardware and software improvements to the original version of the SPS-41. These improvements include dual-port memory option and availability of a double-precision autocorrelation routine.

We are continuing to disseminate information as it becomes available from the manufacturers, particularly SPS, and to exchange information with the other sites in order to avoid duplication of effort and ensure as much compatibility as possible.

REFERENCES

1. N. Kitawaki and F. Itakura, "Nonlinear Coding of PARCOR coefficients", Meeting of the Acoustic Society of Japan, Oct. 1973, pp. 449-450 (In Japanese).

2. J. Makhoul and R. Viswanathan, "Quantization Properties of Transmission Parameters in Linear Predictive Systems", BBN Report No. 2800, April 1974.

3. R. Viswanathan and J. Makhoul, "Transmission Parameters for Linear Predictive Systems and their Quantization Properties", Presented at the 1974 Arden House Workshop on Digital Signal Processing, Harriman, NY, Jan. 14-17, 1974.